Prospects of using Deep Neural Network as alternative to Decision Trees for Combinatorial Background Reduction in Decay Reconstitution

FYS-STK3155 - Project 3

Alexander Umansky
Norwegian University of Science and Technology, University of Oslo
(Dated: 18.12.2025)

During the search for new resonances in the 3-body haronic decay of the B meson $B^- \to \bar{p}\Lambda D^0$, sparsity of the signal after background processing prevented any sound conclusion to be made about potentially new resonances. In this paper we investigate prospects of improving the data processing pipeline, before appending more event data. More precisely, how using Deep Neural Networks and more sophisticated Ensemble methods for Decision Trees can reduce combinatorial background without excessive loss of signal. The analysis considered Adaptive and Gradient Boosted decision trees with Bagging, and multiple architectures of DNNs. This paper presents a selection of classification configurations, each emphasizing different trade-offs between signal preservation and background suppression. Without requiring excessive model depth and compromising model stability, DNNs were shown to preserve more signal albeit with a larger background residual.

I. INTRODUCTION

Deep Neural Networks (DNN) have received great recognition for their universal applicability to complex functional input-output mappings. The layered network of interconnected nodes provides a geometric representation of a trainable model as a *decision surface*. Managing network architecture and optimizing trigger parameters (weights and biases) that control information flow between nodes, effectively reshapes the model in response to provided data. We refer the reader to [1][2][3] for more details regarding signal propagation and training algorithms for DNNs.

The other deep learning strategy we consider is the use of Decision Trees. A tree describes a nested sequence of binary choices (e.g. selection cuts) which partition input data into increasingly homogeneous subsets, that eventually determine an entry on the signal to background spectrum. Singularly, trees exhibit large prediction variances, hence one always uses so called *Ensemble* methods which are further discussed in section II C 1.

Decision Trees find a more traditional use in decay reconstruction for High Energy Physics. Their practical appeal lies in their minimal requirements on data preprocessing. Unlike DNNs, decision trees work without normalization and handle heterogeneous variable types i.e. geometric and kinematic *descriptors*. In the earlier stages of the analysis pipeline, extensive data processing with basic cuts, is undesirable as it can easily jeopardise the signal. However, this method has a higher risk of overfitting even with various ensemble safeguards.

To this end, we compare the qualitative advantages both strategies yield towards a background suppression with minimal losses to signal efficiency. Intermediary, we note computational resources as well as the demanded architectural complexity and amount of training data to achieve adequate performance.

II. THEORY AND METHOD

A. 3-body decay of the B-meson and decay reconstruction at LHCb

The analysis concerns a 3-body hadronic decay of the B-meson at the LHCb detector. The heavy meson ($m_B \sim 5.28$GeV) offers a wide phase space to form intermediate resonances with heavier flavours before the final states $(\bar{p}, \Lambda, D)$ are created. The decay is given by:

$$\begin{aligned} B^- &\to \bar{p} D^0 \Lambda \\ &\hookrightarrow \Lambda \to p\pi^- \\ &\hookrightarrow D^0 \to K^-\pi^+ / K^-\pi^+\pi^+\pi^- \text{ (D2/4H)} \end{aligned} \tag{1}$$

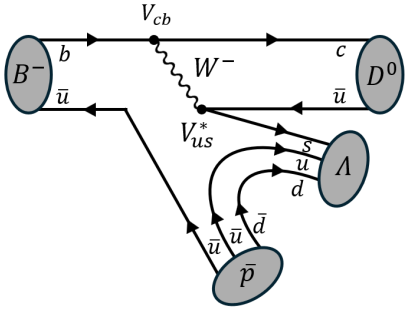


Figure 1: Lowest-order Feynman diagram for $B^- \to \bar{p}\,\Lambda\,D^0$. The b quark inside the B meson decays weakly to a c quark, and W^- transforms into a $\bar{u}s$ pair. The s quark goes to form the Λ baryon (uds) while c hadronizes into a D meson ($c\bar{u}$).

the b quark inside B meson decays via the weak interaction into c (and s from the W) quarks that will later hadronize into a D meson and Λ baryon respectively. The process offers a novel insight into intermediate charmed and strange $\Xi_c^0(csd)$ baryonic resonances, new excited kaon $K(s\bar{u})$ states, and possibly even exotic

hadrons such as pentaquarks $P(\bar{u}u\bar{d}\bar{u}c)$. We then consider Λ decaying to a pion and proton, and D decaying to a kaon and either one or three pions as shown in eq.1. From now on we label the D's decay channels as *D2H* and *D4H* respectively.

The LHCb is a detection station built to track charged particles $(\pi^{\pm}, K^{-}\bar{p}, p)$. The first element of LHCb is the Vertex Locator (VELO), that pinpoints the origin of a charged track. Particle trajectory is recorded by the Upstream and Downstream trackers (UT, SciFi). A track is called Long (L) if it has been registered by all 3 stations. Downstream tracks (D) are not registered by VELO, likely stemming from a long-lived particle like Λ. In between, a magnet is used to extract particle momenta from the curving trajectory. We consider its *up/down* polarization to account for systematic errors. Using momenta, vertex positions, and trajectories of the *grand-daughter* particles, we can trace back the decay and reconstruct the *daughters* (D^0, Λ) and the initial particle (B^-). Intuitively we expect Long tracks to be the most accurate, and the *D4H* channel to have worst kinematic resolution.

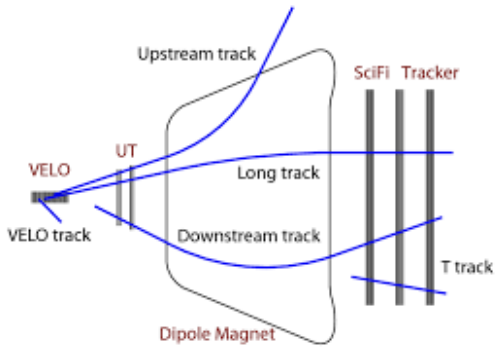


Figure 2: A schema of the LHCb detector showing the VELO, magnet, and the Uprestream and SciFi trackers. The blue lines represent (in our cause) the tracks left by B's *charged* decay products $(\pi^{\pm}, K^{-}, p, \bar{p})$.

B. Preselection and Truthmach

The first step of the analysis is removing obvious background by constraining kinematics and vertex quality to impose momentum conservation and ensure candidate tracks trace back to the *Primary Vertex* and are not miss identified. Examples of types of cuts are:
- P_T and P (LHC environment is highly boosted)
- Impact Parameter and Direction Angle χ^2s
- Particle miss-identification

The selection cuts are intentionally kept loose so as to not lose valuable signal. The remaining background region will be used to train the classifiers. As already mentioned, each stage of signal processing aims to *surgically* remove a fraction of the background. Besides preprocessing and combinatorial analysis, the are also amplitude analysis and S-fit which we do not discuss here. In later stages we obtain more sophisticated selection descriptors that preserve signal efficiency far better. Further details on preselection cuts can be seen in A1.

Parallel to event data, we use a Monte Carlo (MC) simulated decay. It will be used as proxy for "true signal" during classification training, and to evaluate efficiency of background removal (see section [*]). For the MC sample a *Truthmatch* is needed to enforce particle identity and parentage in the simulated decay using the generator-level labels. Note that for MC and event samples individual treatment will apply depending on channel, track, and less critical polarization and year. TableI shows the efficiency of truth-matching, and Fig3 shows the reconstructed B-mass after the selection cuts were applied combined for track, year and polarization. As expected, *D4H* with two additional pion tracks has worse resolution around the peak 5.28GeV. We purposefully set B_M $>$ 5.1GeV during preselection to cultivate a large background region which we call the *upper band*. The non-Gaussian upper sideband implied presence of combinatorial background (the mothers have been assigned wrong children). The values B_M $>$ 5.5GeV will be used as background test sample in the upcoming classification.

Channel	Track, Pol*	Total	Preselected	Efficiency (%)
D2H	*DD* ↓	49 691	20 175	40.60 ± 0.22
	DD ↑	50 298	20 400	40.56 ± 0.22
	LL ↓	13 457	5458	40.56 ± 0.42
	LL ↑	13 199	5340	40.46 ± 0.43
D4H	*DD* ↓	13 294	4385	32.98 ± 0.41
	DD ↑	12 536	4119	32.86 ± 0.42
	LL ↓	3779	1115	29.51 ± 0.74
	LL ↑	3541	1083	30.58 ± 0.77

Table I: Truthmatch cuts for channel, polarization and track cuts based on TRUEID and MOTHERID. Private MC samples for 2016 data (only), *Magnet Polarization MU↑/MD↓.

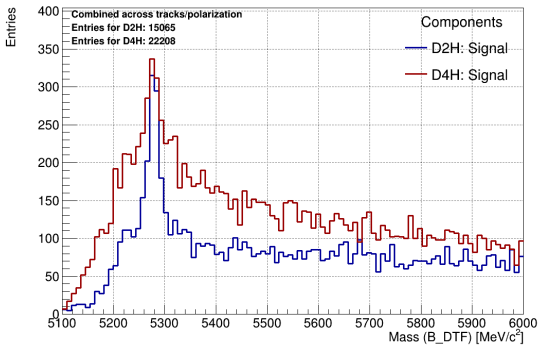


Figure 3: The B invariant mass combined over year, track and polarization. Notice that we have much less entries for *D2H*. Channel *D4H* has 2 more pion tracks to constrain so its reconstruction resolution is worse. The upper sideband of B inv. mass has signature of combinatorial background. It will be used as a background training sample.

C. Background Classification

To account for experimental factors, 4 separate classifiers will be trained for each track and channel combination. The truth-matched MC-data, and the upper sideband $m_B \in (5.6, 7)$GeV data-region (merged over year and polarization) will serve as signal (S) and background (B) samples respectively. For each, events are split randomly for testing and training, while event-weight will be normalized to the sample size: $w_{S,B} = 1/N_{S,B}$.

The discriminating variables will be the vertex geometry of B, D, and the momentum of the final state products: $\bar{p}, p, \pi_n; n = 1, (2, 3$ if D4H). The discriminators were chosen by principles of reconstruction logic; At this stage, using mostly track-level observables as discriminators is preferred to a large number of features, of which any fit-based, will introduce reconstruction-bias. Then assuming approximately Gaussian-like distributions for undistorted observables, a logarithmic transformation was applied to reduce distribution overlaps.

log Classification Variable	
B_IPCHI2_OWNPV	
B_ENDVERTEX_CHI2	
D0_ENDVERTEX_CHI2	
D0_FDCHI2_ORIVX	
D0_K_IPCHI2_OWNPV	
D0_pi(n)_IPCHI2_OWNPV	
D0_pi(n)_PT	
p_PT	
Lambda_p_PT	$\}$ MAX($p_T^{\Lambda_\pi}$, $p_T^{\Lambda_p}$)
Lambda_pi_PT	
train BKG: B_M $\in (5600, 7000)$(*because of LL)	

Table II: Discriminators to be used in classification training. Variables marked with PT are momenta of final state particles. Track-level observables are not biased by reconstruction and vertex-constraining. Assuming Gaussian distributions, log-scaling was performed for each discriminator to separate overlapping regions.

1. Classification Models and Ensemble Methods

The *paradigm* of Ensemble methods is to train large number trees independently towards reducing prediction variance which is inherently large for decision trees. The following properties will remain fixed: maximal depth (number of decision nodes) of any tree is set to 3; Adding more *leaf* nodes will either increase risk of overtraining or require more computationally demanding decorelative measures. Descriptor variables are assigned with *cuts* to further granulate the feature space. Thus a decision node f.ex: X_PT $\in v_x \subset \mathbf{V}_x$, leads to a decision spectrum for a binary classifier. The number of cuts must be adequate compared to the total number of trees and random feature sections $n_{cuts} \ll N_{trees}$. Node splitting criterion to be used will be *Cross Entropy* or Gini index: both describes classification impurity to be minimized.

$$G = 1 - \sum_{c=S,B} p_c^2, \quad H = \sum_{c=S,B} -p_c \log(p_c) \qquad (2)$$

For more information regarding how to implement ensemble methods for event data, we refer to [3][4][5][6].

i. **Bagging** (bootstrap aggregating) methods train *independent* trees on resampled (with replacement) subsets of the training data and then aggregate their predictions (either by averaging or majority). Bagging is implemented to reduce high variance for training sensitive to input-perturbation. It is however not capable of improving bias inflicted data.

ii. **Random Forests** extend bagging; additionally decorrelating trees through random feature selection at each split. This reduces the correlation between individual trees, leading to a further variance reduction (saturates past certain point[3]). Random forest are much more resilient to overtraining, provided that number of built trees is adequately large, and is not overly correlated.

iii. **Boosting** is a bias-reducing method. Unlike Bagging and RF which train in parallel, Boosting builds the discussion trees *sequentially*, combining weak classifiers into a stronger by emphasizing events that were misclassified in previous iterations. *Adaptive* boosting reweights misclassified samples, while *Gradient* boosting optimizes a differentiable loss function by fitting successive trees to the *residuals* of earlier models.

Unlike BDT, DNN classifiers require feature *standardization*. For kinematic variables and vertex geometry we apply Gaussian standardization (G) [4]. We explore a similar depth of 2-3 hidden layers, and "funnel" shaped node structure $n^l \sim 2^{L-l}$. $L1$ regularization $\lambda \sim 10^{-4}$ was added a safety measure against overfitting. Optimization is based on the famously robust $ADAM$ method as noted in previous studies[1]. Finally, we used mini-batch method with batch size 64 to balance convergence speed and classification performance.

2. Evaluation

The classifier assigns to each event a response value y, defined either as a continuous score or as a discretized output depending on the decision model (BDT or DNN). This response represents the projection of the input feature space onto a one-dimensional discriminant optimized for signal-background separation. The performance and

stability of the classifier are evaluated using distribution and threshold-based metrics. The *Kolmogorov-Smirnov* (KS) test is used to evaluate fidelity between training and testing output distributions, thus diagnosing overtraining. Defining $F_c(y)$ as the cumulative distribution function of the response for class $c = S, B$, and taking the supremum (mean distance) over y:

$$D_{\mathrm{KS}} = \sup_y |F_S(y) - F_B(y)| \tag{3}$$

where $D_{\mathrm{KS}} \to 1$ describes the probability of fidelity. Although, larger value is preferable, a generally more realistic expectation for high energy physics are values above the threshold: $D_{\mathrm{KS}} > 0.05(5\%)$.

The other diagnostic is the *Receiver Operating Characteristic* (ROC) curve. For a given threshold y_{cut}, the ROC curve shows the trade-off between signal and background efficiency as function of the threshold:

$$\epsilon_c(y_{\mathrm{cut}}) = \int_{y_{\mathrm{cut}}}^{y_{\max}} P_c(y)\, dy, \quad c = S, B \tag{4}$$

Intuitively the goal is to achieve maximal efficiency for both, quantified either by *Area Under the Curve* (AUC $\to 1$), or a more rigorous metric: the Figure of Merit.

3. *Figure of Merit Scan*

For continuous classification response variable y we perform a separate *Figure of Merit* (FOM) scan. Defining **S0', B0'** as the proxy signal and sideband background yields, and use **S0, B0** to label the signal and background components of the resonance peak ~ 5.3GeV (see Fig3). Using constant: $fs = S0/S0'$, $fb = B0/B0'$, then loop through BDT cuts and update $S0', B0'$. Recalculate: $S = S0' \cdot fs$, $B = B0' \cdot fb$ and determine the FOM based on Posson statistics for events:

$$FOM = \frac{S}{\sqrt{S+B}} \cdot \frac{S}{S+B} \tag{5}$$

Optimization of y_{cut} is performed though the maximization of the FOM. This is a more advantageous method, leveraging both the yield within the regions used to train the classifier and the specific resonance region, thus more aligned with the concrete analysis objective. After applying the background filter, we look for duplicate entries for each event. One is chosen for each run/event category. Efficiency is catalogued using binominal statistics: N_0, N' for initial and surviving events.

$$\varepsilon = \frac{N'}{N_0}, \quad \sigma_\varepsilon = \sqrt{\frac{\varepsilon(1-\varepsilon)}{N_0}} \tag{6}$$

Data Regions
Polluted SR $\in$ SIG **S0**, BKG **B0**, Range: 5300 MeV $\pm$ 40 MeV
Extended Sideband SB $\in$ BKG **B0'**, Range: (5500, 7000)MeV
MC Region
Proxy SR $\in$ SIG **S0'**, Range: (5300, 7000)MeV
Scan Configuration
Additional D^0 mass cut: $\lvert m_{D^0} - 1864.84\rvert < 50$ MeV
SR width (B_0, S_0): 40 MeV
SB definition (B_0'): [5500, 7000] MeV
Initial BDT cut: -0.4
FoM scan for BDT values: $-0.4, -0.2, \ldots, 0.2$ (step 0.01)

Table III: Defining the sideband (SB) and Signal regions (SR) for event and proxy data. The FOM scan conflagrations for BDT output y are shown.

D. Resources and Acknowledgements

The classification analyses uses the TVMA Roottool[5] which was designed to work with reconstruction event-data. The tool supports other classifiers, besides BDTs. The DNN model uses the Feedforward and Backgpropagation algorithms, allowing all necessary but basic customization of network architecture. Note this option will be deprecated in later updates.

We acknowledge the contribution of Song, Y. and Ding, Y. at (EPFL-LPHE) for providing the *private* Monte Carlo simulation data for 2016.

Preselected data and Truthmatched MC, along with the classification and evaluation scripts will be accessible here: https://github.com/4Lexium/Data-Analysis-and-Machine-Learning/tree/main/project3.

We also made active use of OpenAI: ChatGPT and DeepSeek to suggest architectures for BDTs and DNNs.

III. RESULTS AND DISCUSSION

A. Correlations

Fig.4 shows the pairwise correlations among the discriminative variables plotted for D2H, D4H and signal, background. Note the difference based on tracks was insignificant. Overall, the selected kinematic and geometric descriptors yield a presentable correlation structure. Exception being momenta of Λ's decay products which introduce a redundancy, so one can be removed. The D4H channel has more decay products and displays stronger correlation among its geometric properties. Nevertheless, these correlations remain below alarming levels of multicollinearity. Importantly, neither signal nor background exhibit systematically poorer representation by the chosen descriptors for either channel. This means the classier will not be intrinsically biased.

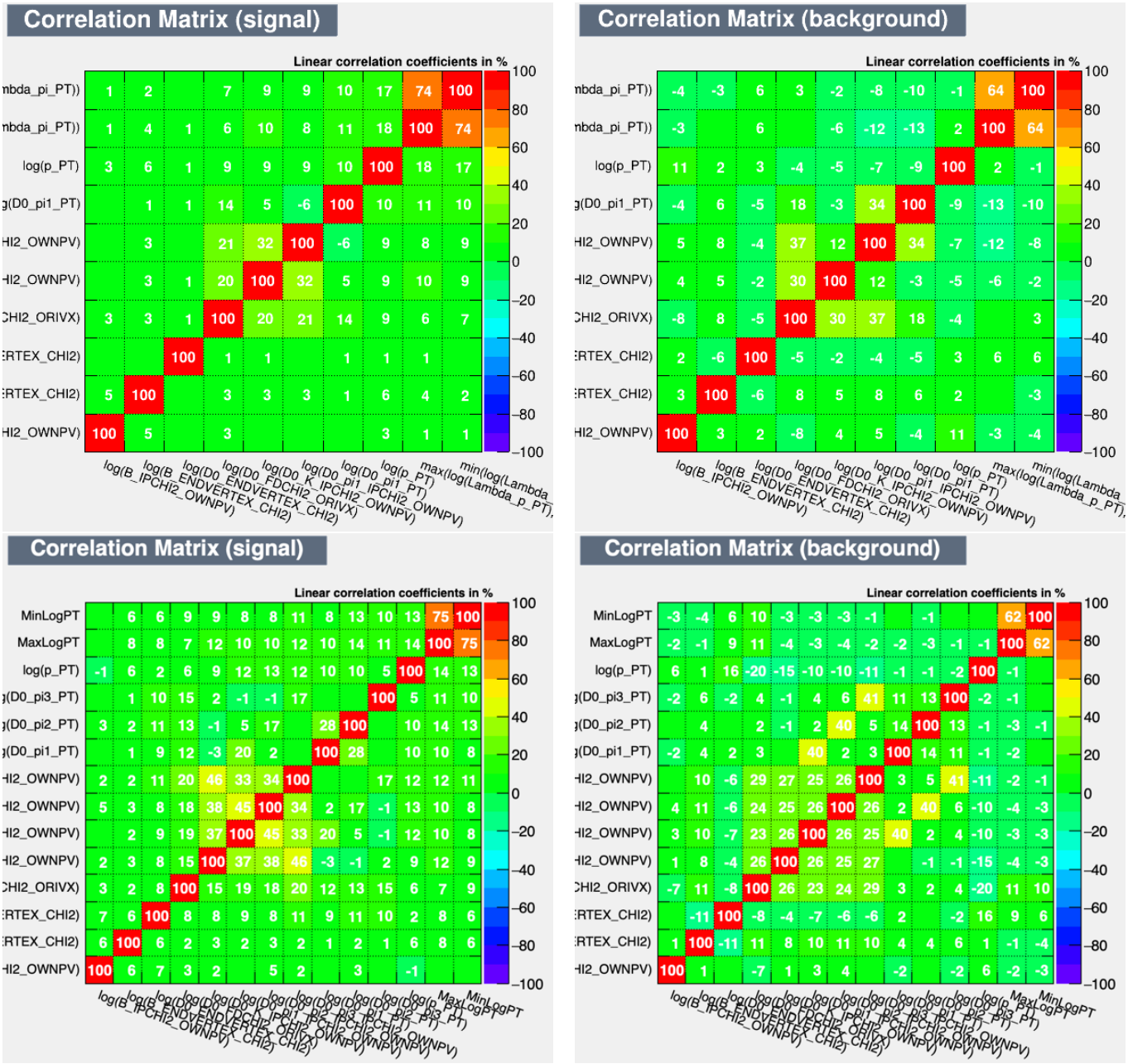


Figure 4: Top Row: D2H_DD, Bottom Row: D4H_DD. Figures display the pairwsie correlation structure for kinematic and geometric descriptors. Overall verdict is that the variable selection is good (note the policy of using tracking-observables with minimal reconstruction bias). The decay products of $\Lambda \to p, \pi$ are strongly correlated. We attempted to use $\min/\max(p_\pi, p_p)$, but this did not resolve the dependency. Hence, one variable is sufficient. D4H channel has many more decay products with correlation among the vertex-quality variables. Neither of these correlations are critical, the disruptors can be kept without risk of multicollinearity. The correlation profile changes insignificantly based on tracks. Finally, we observe no significant change to correlation structure for signal or background given the set of descriptors.

B. BTD and DNN Classification

Four independent classifiers are trained with different input samples. Recall from earlier the experimental difference between Long and Downstream tracks, and the resolution quality for D2H and D4H channels. In this analysis, we fix *one* mass-sideband for all classifiers, and instead adjust this interval to the worst learner: LL. The outcome of using a common sideband is the sample size imbalance for the background training data:

$$\text{D4H\_DD} : 7850, \quad \text{D4H\_DD} : 13400$$
$$\text{D4H\_LL} : 3800, \quad \text{D2H\_LL} : 1861$$

Although, statistical sparsity can be mitigated through Bootstrapping, resampling does not improve deficiencies or limited descriptive information in the data. Hence, expanding the sideband was seen to be more effective and was done until background classification reached minimal thresholds $\sim 5\%$ for the LL. The sideband is constrained up to 7GeV to maintain a balance between background-positive and background-negative sample sizes. Note also that no explicit cut was placed on the resonance peak; The classifiers learn to distinguish the combinatorial background and not the signal.

In section II C 1 we mentioned Random Forests. For boosted decision trees in TVMA, this method is not directly implementable. However, with Bagging enabled for a large number of independent trees with granular selection cuts imposed by *nCuts* parameter, the methods yield an improved robustness against overtraining and decorrelation of decision trees.

1. BDT: Adaboost

Adaboost is a reweighting-based algorithm, hence the effective response distribution retains the Gaussian profile perturbed by individual classifications. Large number of spikes and sharp drops indicates that the classifier was overtrained and learned noise. This is further emphasised by overlaid train/test distributions and the KS probability in Fig5. The best and worst fidelity is observed for the D2H_DD and D2H_LL channels respectively. For LL classifiers, with limited background statistic, we applied test configuration B with reduced granularity *nCut* and boosting momentum *beta*. This nearly doubled the KS values for D4H_LL and brought D2H_LL to 2% threshold. Overall, using 600 construed trees, and no more than 1 leaf node was seen to reduce overtraining. Lesser improvement include: using Cross Entropy H instead of Gini, *nCuts*: $20 - 100$ and Bagging fraction: $0.5 - 0.7$.

The overlap between background and signal distributions, means an optimal BDT cut is not evident by inspection, and must be evaluated based on the FOM scan. The ROC curves and AUC for Adaboost describe good signal and background efficiency, even for the LL classifiers. We regard this statistic with caution and use the AUC value as one of the benchmarks to evaluate improved or degraded classification performance.

2. BDT: Gradboost

Gradient boosted BDTs produce qualitative different repose shape characterized by a triangular slope, reflecting the sequential procedure: $F_{i+1}(y) = F_i(y) + \eta R_i(y)$, where η is controlled by the *shrinkage* paramter and R is the misclassification residual. Fig.6 depicts the first attempt of naively reusing the same configurations as for Adaboost. It is evident from the degree of overlap, and large variance between train and test samples that *Test 1* is failed. Towards Fig7 we use a lower shrinkage parameter $\eta : 0.1 \rightarrow 0.001$ together with lower *nCuts* and higher bagging fraction 0.7. These adjustments were seen to significantly improve the ROC efficiency and the KS overtraining metric. Indeed, the LL channels experienced the biggest relative improvement.

Gradboost method produced a more compact response distribution as evidenced by a narrow band $y \in (-4, 4)$. Despite the overlapping region seemingly improving, it is important to note that the classifier assigns larger fraction of events to extreme scores. The loss of discriminative granulation means classification categorizes more brutally and the decision surface becomes less informed.

3. DNN

Input features were standardized using an built-in TVMA method. We recommend *Var Transform*: G, which normalizes to zero mean and unit variance. The network architecture was designed using *ReLu* activation functions for hidden layers and a *Sigmoid* for the output layer. This binary classification provides an adjacent theory to the distribution of Adaboost-BDT response. The amount of misclassification depends on how well the DNN can interpret the underlying pattern and ignore outliers and noise. Several configuration of hidden layers and node structure (constituting model complexity) were tested, see Fig8 Test A, B. Adding more hidden layers than $2-3$ is not advised, as this only stimulates the network to learn more fluctuations. Instead a wider funnel-network was seen to yield *saturating* improvement. We note additional improvements: *TestRepetitions* builds and optimizes multiple network models, then selects the best performer. Optimization was done using *ADAM* in mini-batch mode with batch sizes 64. Learning rate serves as a protective measure against overtraining, but depending on TMVA standardization, it may offset features differently, $\lambda < 10^{-3}$ are advised.

DNN response, seen in Fig8, further underlines the idea of misleading classification distributions for Gradient BDTs. Due to the sigmoid in the output layer, events are assigned to extreme edge cases $(0, 1)$. As evidenced by amount of background misclassified as signal, the Net-

work is not capable of fully interpreting the distinction. Increasing the network complexity did not remove this phenomenon entirely. Unlike for Adaboost and Gradboost, the DNN misclassification is also barbarized, and no response cut can remove the misclassified region.

C. FOM Scan

For Adaboost classification result, FOM scan was employed following the previously described routine (II C 3). We summarize the cut efficiency in Table IV and plot the filtered resonance region in Fig.9. Note how the efficiency drops for D2H_DD and D4H_DD. Shorter downstream tracks *D* exhibit stronger combinatorial background, and although the *Adaboost* BDT cut achieves an impressive background suppression, this comes at the cost of signal loss, almost double that sustained by *L* tacks.

Channel	Track	Post-BDT cut	Post-Selection	Efficiency
D2H	DD	597	596	0.046 ± 0.002
	LL	792	785	0.215 ± 0.007
D4H	DD	318	310	0.017 ± 0.001
	LL	807	778	0.077 ± 0.003

Table IV: Applying the Adaboost-BDT cut found iteratively though the FOM scan. Then choose randomly one candidate for each run/event category. The efficiency of these combined selections is calculated using eq.6.

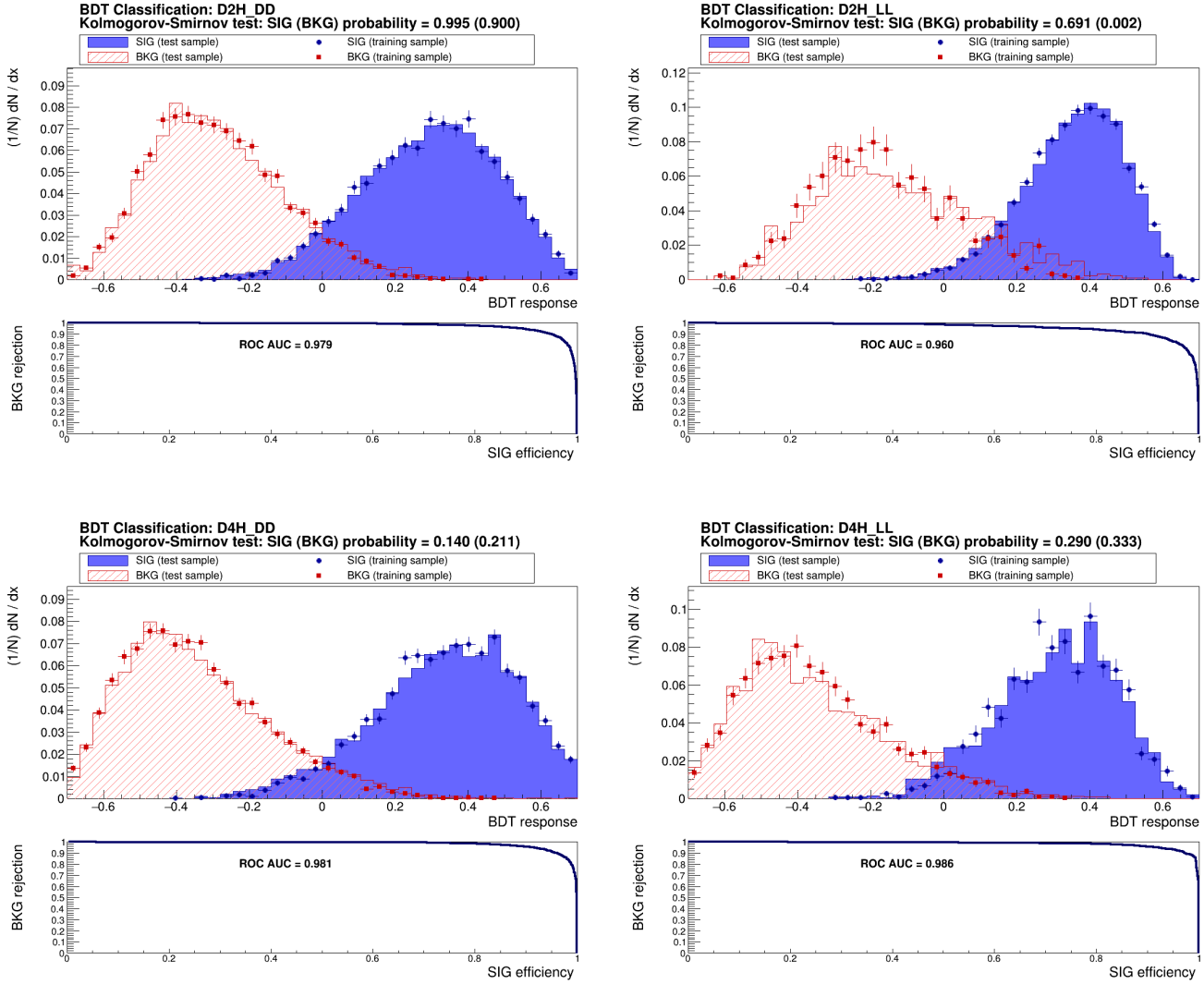
Fig.10 presents the results of using a DNN classifier with architecture A, and a subsequent cut $y > 0.8$. The classifiers were observed to struggle less with *DD* tracks, and in case of DNN, there seems to be far less misclassification in the region close to $y \sim 1$. While this model filter is less effective at removing all the background compared to Adaptive BDT, it preserves a significantly larger fraction of the signal peak. Further gains in efficiency were achieved by adopting a wider network architecture (B), as demonstrated in Fig.11(left). Increasing the network depth or using a larger number of nodes per layer, did not yield an improvement. Past some architectural complexity the network only learns more noise and not the combinatorial pattern.

Fig.11(right) is the result of using *Gradboost* BDT (Test 2) with cut $y > 0.1$. The performance places between that of DNN and Adaboost: relative to the DNN, less signal is retained, while background suppression is more effective; relative to Adaboost, the signal loss is reduced at the expense of weaker background rejection.

IV. CONCLUSION

The primary objective of this study was to evaluate alternative classification methods with the intention of maximizing signal preservation. Given that the full analysis pipeline applies multiple successive filtering stages, a comparatively weaker suppression of combinatorial background at the classification stage can be partially compensated by f.ex. the preceding SFit procedure. With this in mind, both DNN model (B) and Grading Boosted decision trees (2) prove to be competitive and merit further consideration.

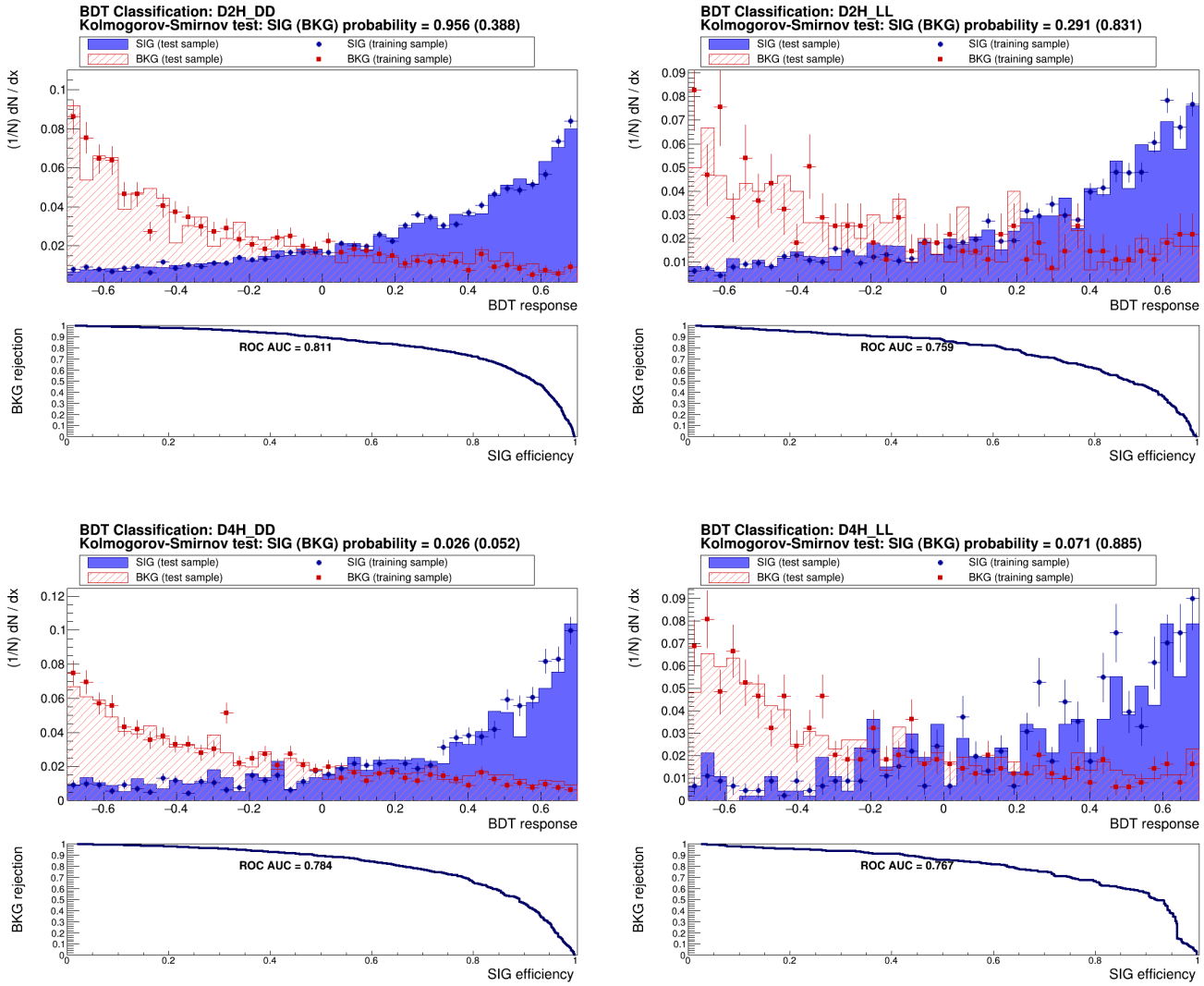
It is worth considering the adoption of more advanced TensorFlow-based network tools and prospects of using Extreme Gradient Boosting (XGB) and Random Forests. These implementations offer improved control of overtraining, likely enabling probing with deeper networks or decision trees towards further optimization of the said signal-background trade-off.



BDT: Test AdaBoost A(B)

`SplitMode: Random, NormMode: NumEvents`
`SIGCut: None, BKGCut:` $B_M \in (5600, 7000)$
`NTrees: 600, MaxDepth: 3, nCuts: 200(100)`
`BoostType: AdaBoost, BoostBeta: 0.3(0.15)`
`BaggedSampleFraction: 0.5()`
`SeparationType=CrossEntropy`

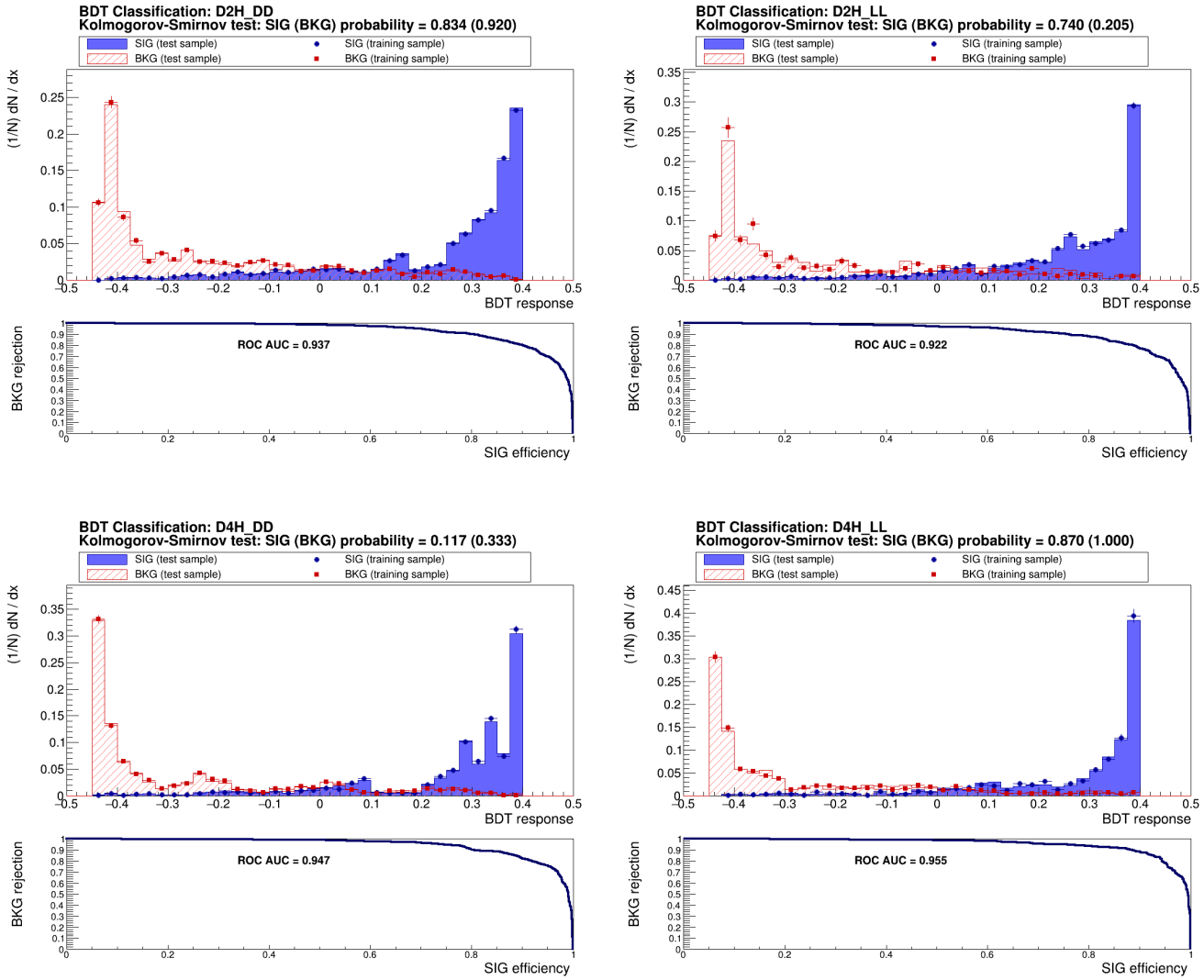
Figure 5: Adaptive boost response distributions for signal and background. Sharp spikes and drops indicate residual overtraining, confirmed by train-test comparison and the KS probability. Worst performance is recorded for LL methods which have fewer background entries. These cases required different classification configurations (B). Lowered boost value and nCuts were seen to improve the KS metric past 2% confidence for the worst classier D2H_LL. Overall the efficiency curves and distributions show promise. The optimal BDT cut will be evaluated using a FOM scan.



BDT: Test 1 GradBoost

```
SplitMode:  Random, NormMode:  NumEvents
SIGCut:  None, BKGCut:  B_M ∈ (5600, 7000)
NTrees:  600, MaxDepth:  3, nCuts:  50
BoostType:  GradBoost, Shrinkage:  0.1
BaggedSampleFraction:  0.7
SeparationType=CrossEntropy
```

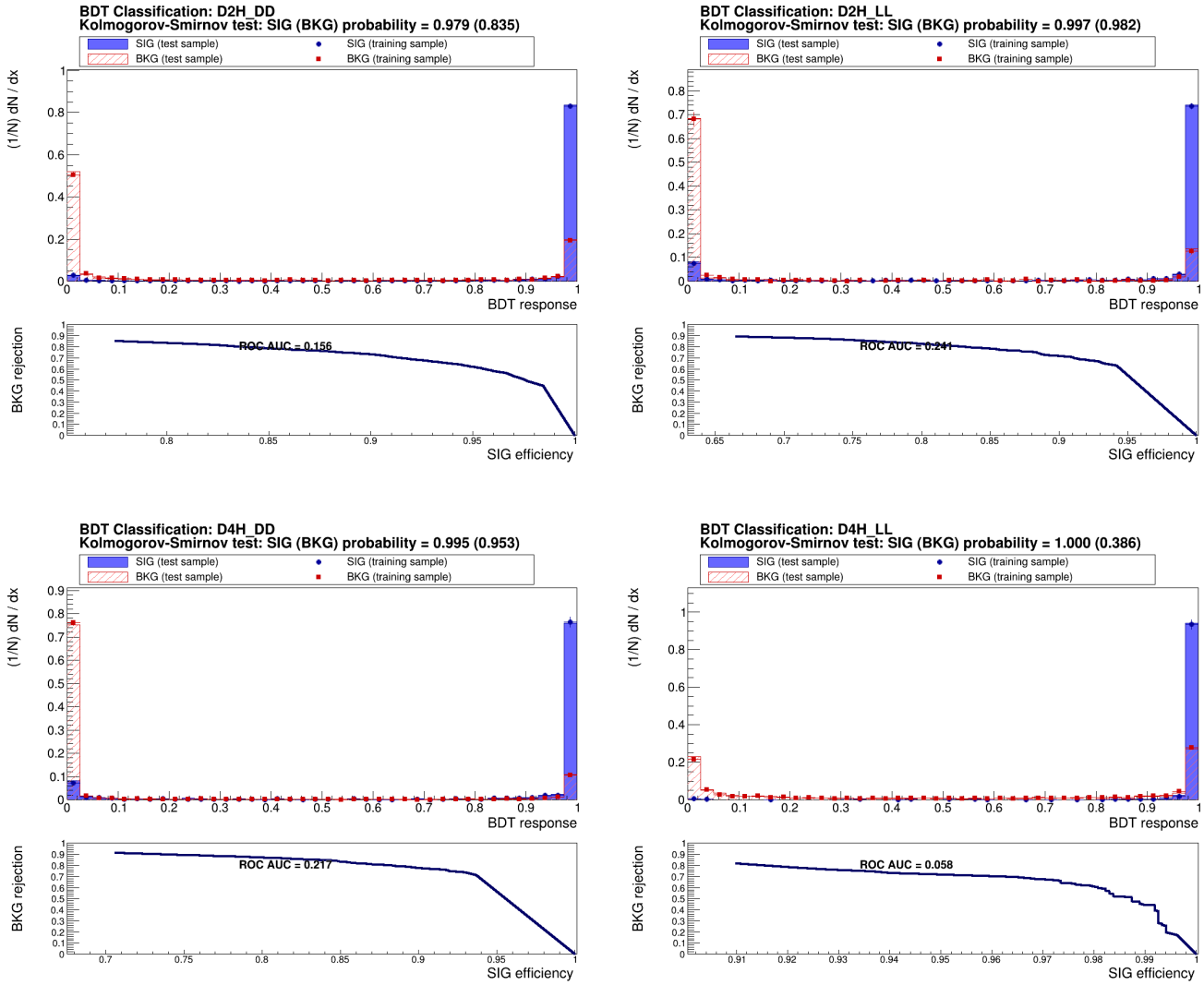
Figure 6: Gradient Boosting produces a characteristic triangular response due to its sequential residual fitting. Reusing AdaBoost parameters (Test 1) results in large train–test discrepancies and poor KS statistic. Also the ROC curve and AUC performance degrades the signal/background efficiencies.



BDT: Test 2 GradBoost

```
SplitMode: Random, NormMode: NumEvents
SIGCut: None, BKGCut: B_M ∈ (5600, 7000)
NTrees: 600, MaxDepth: 3, nCuts: 20
BoostType: GradBoost, Shrinkage: 0.001
BaggedSampleFraction: 0.7
SeparationType=CrossEntropy
```

Figure 7: Second test with GradBoost. Using a lower shrinkage parameter $\eta = 0.001$, reducing the cut value and increasing the bagging fraction significantly improved the ROC performance. Furthermore, the KS overtraining metric is presentable, with an impressive relative improvement for LL channels. However we note that the resulting compact response concentrates events at extreme scores, reducing discriminative granularity and yielding a more aggressive, less informative decision surface.



DNN: Test Architecture A(B)

SplitMode: Random, NormMode: NumEvents
SIGCut: None, BKGCut: $B_M \in (5600, 7000)$
VarTransform: G, ErrorStrategy: CROSSENTROPY
Architecture (A): RELU|128,RELU|64,RELU|32,Sigmoid
Architecture (B): RELU|256,RELU|128,RELU|64,Sigmoid
Optimizer: ADAM, MaxEpochs: 200, BatchSize=64
LearningRate: 1e-4, Momentum: 0.9
TestRepetitions: 5

Figure 8: DNN inputs are standardized to zero mean and unit variance and trained using ReLU activations with a sigmoid output. Performance saturates for architectures beyond two to three hidden layers; wider networks are preferred over deeper ones. Despite optimization with ADAM and small learning rates, the sigmoid output forces predictions toward extreme values, producing polarized misclassification. Unlike BDTs, this misclassification cannot be mitigated by response cuts, indicating limited interpretability of ambiguous regions despite visually clean separation.

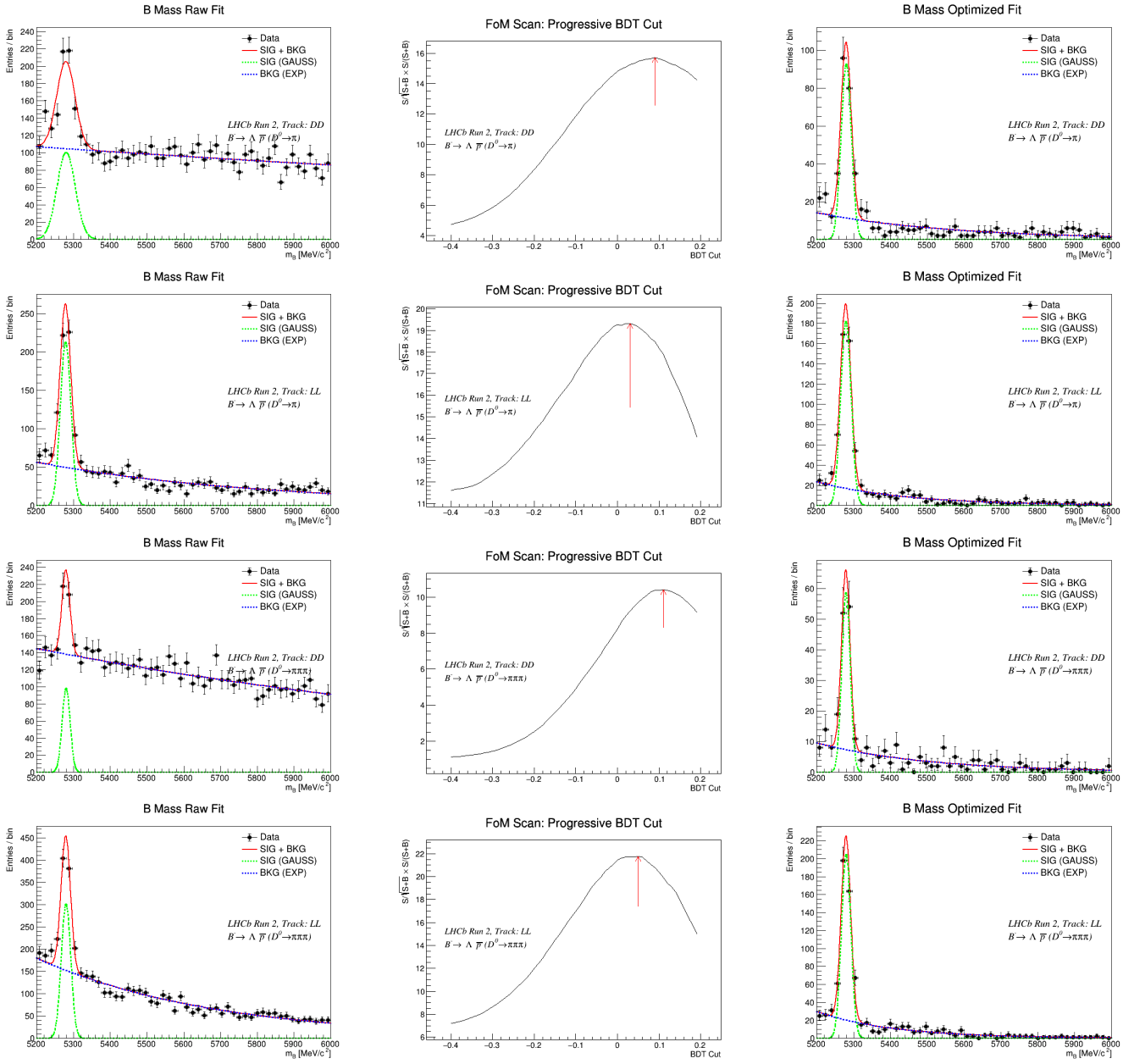


Figure 9: Figure of Merit (FOM) scans for different decay channels. Each row represents a specific channel: (row 1) D2H_DD, (row 2) D2H_LL, (row 3) D4H_DD, and (row 4) D4H_LL. Columns from left to right show: raw fit results, FOM scan profiles, and optimized fit results for each channel. The optimal *Adaboost* BDT cut corresponds to the maximum of the FOM curve. The DD data was seen to have strongest combinatorial background (shorter tracks in the detector). Following the classification cut, the combinatorial background is well suppressed, albeit with noticeable signal reduction for the DD tracks. The multivariable fit was performed using the ROOFIT libary tool [7]. The signal component was modelled using a Gaussian and the background as an exponential.

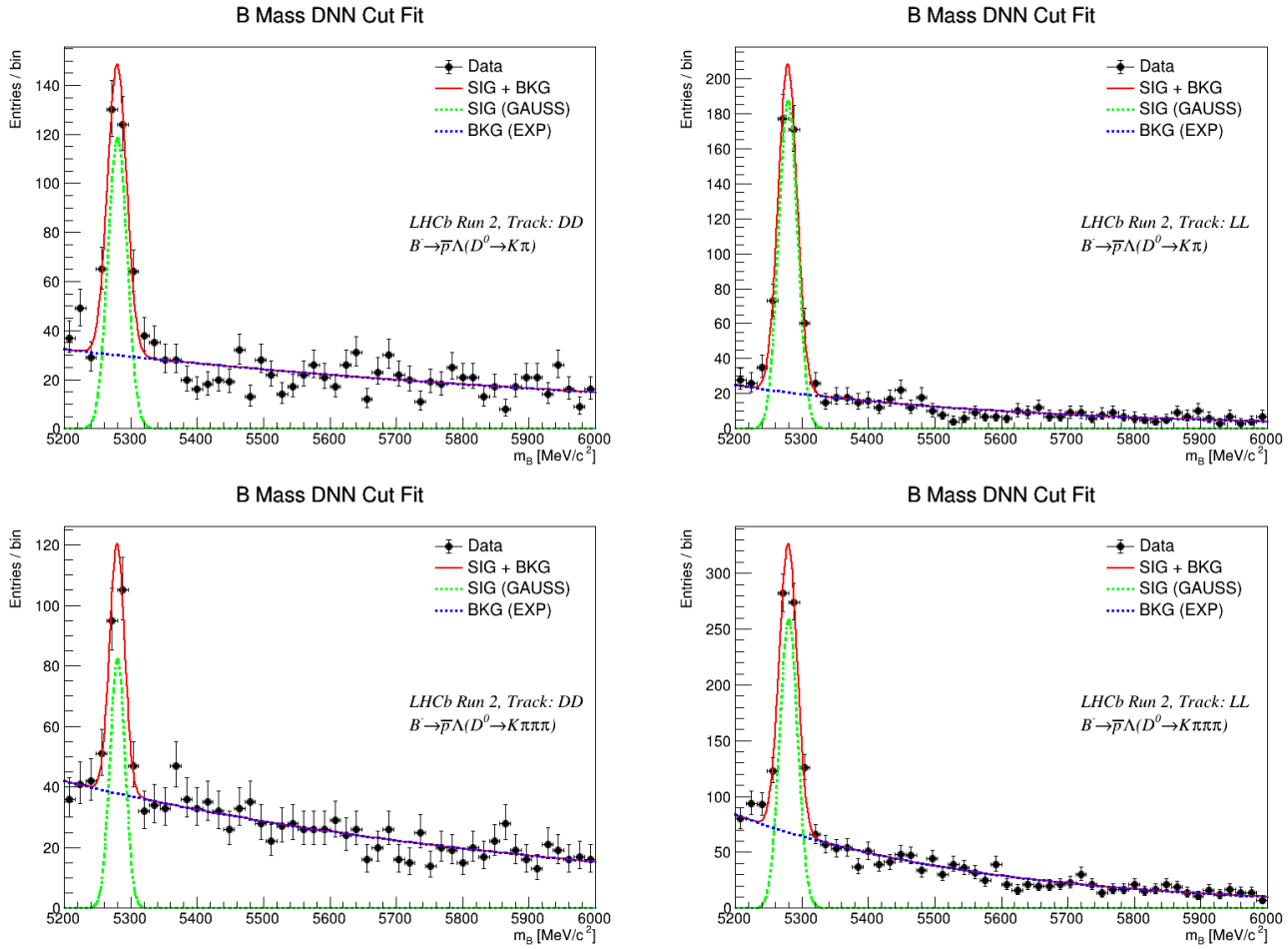


Figure 10: DNN model of Architecture A, filtered using cut $y > 0.8$. Despite the misclassifications observed during classifier evaluation, we observe a batter preservation of the signal. This is especially relevant for the DD channels, where the DNN has fewest recorded missclassifications. The background is suppressed less effectively compared to the BDT methods.

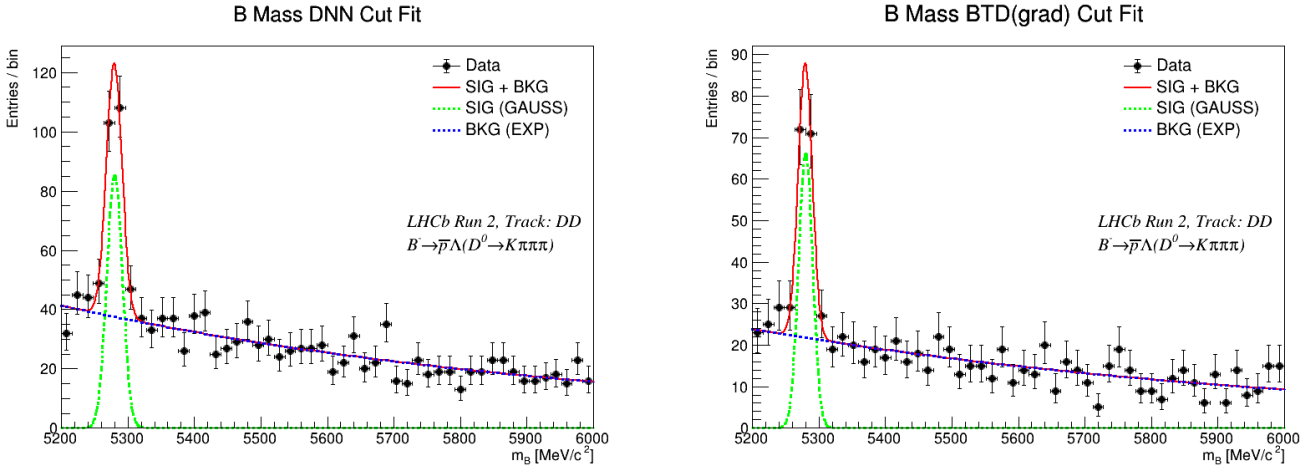


Figure 11: **Left**: Using a wider Network with architecture B. achieves a better efficiency i.e. more signal is preserved and the background is further suppressed. Increasing the depth of the network does not improve the classification, as the network only learns more noise. A wider network (higher node number) yielded a saturating improvement. **Right**: Gradient-boosted BDT (Test 2) using cut $y > 0.1$ for DD4H_DD. The result places inbetween the performances of Adaptive BDT and DNN.

A1. PRESELECTION CUTS

Variable	Selection Criteria
B_PVFit_M	> 0
B_DTF_MASS_2	> 0
B_PVAllFit_M	> 0
B_ENDVERTEX_CHI2/NDOF	< 9
B_IPCHI2_OWNPV	< 25
B_DIRA_OWNPV	> 0.99
B_M	> 5100 MeV

Table V: B Meson Selection Cuts

Variable	Selection Criteria
p_ProbNNp	$p_{\mathrm{ProbNNp}}(1 - p_{\mathrm{ProbNNk}})(1 - p_{\mathrm{ProbNNpi}}) > 0.3$
Lambda_ENDVERTEX_CHI2/NDOF	< 10
Lambda_DIRA_ORIVX	> 0.8
Lambda_FDCHI2_OWNPV	> 25
Lambda_M	$1000 < M < 1200$ MeV
Lambda_p_PT + Lambda_pi_PT	> 1000 MeV
Lambda_pi	$P > 1200$ MeV & ProbNNpi > 0.2
Lambda_p	$P > 1200$ MeV & ProbNNp > 0.3
Lambda_pi_IPCHI2_OWNPV	> 4 (DD) or > 9 (LL)

Table VI: $\bar{p}$ and Λ Selection Cuts

Variable	Selection Criteria
D0_ENDVERTEX_CHI2/NDOF	< 10
D0_FDCHI2_OWNPV	> 36
D0_DIRA_ORIVX	> 0.8
D0_M	$1800 < M < 2000$ MeV
D0_K_PT + D0_pi1_PT	> 1800 MeV
D0_K	$P > 3000$ MeV & ProbNNk > 0.3
D0_pi1	$P > 2000$ MeV & ProbNNpi > 0.2

Table VII: $D \to K\pi$ (D2H) Selection Cuts

Variable	Selection Criteria
D0_ENDVERTEX_CHI2/NDOF	< 10
D0_FDCHI2_OWNPV	> 36
D0_DIRA_ORIVX	> 0.8
D0_M	$1800 < M < 2000$ MeV
D0_K_PT + D0_pi1_PT + D0_pi2_PT + D0_pi3_PT	> 1800 MeV
D0_K	$P > 2500$ MeV & ProbNNk > 0.3
D0_pi1	$P > 2000$ MeV & ProbNNpi > 0.2
D0_pi2	$P > 2000$ MeV & ProbNNpi > 0.2
D0_pi3	$P > 2000$ MeV & ProbNNpi > 0.2

Table VIII: $D \to K\pi\pi\pi$ (D4H) Selection Cuts

[1] A. Umansky, FYS-STK3155 - Project 2 (2025), unpublished project report, University of Oslo, URL https://github.com/4Lexium/Data-Analysis-and-Machine-Learning/tree/main/project2.

[2] I. Goodfellow, Y. Bengio, and A. Courville, in *Deep Learning* (MIT Press, 2016), accessed October 5, 2025, URL https://www.deeplearningbook.org/contents/optimization.html.

[3] M. Hjorth-Jensen, *Computational Physics Lecture Notes 2015* (Department of Physics, University of Oslo, Norway, 2015), week: 37, 41, 42, 43, URL https://compphysics.github.io/MachineLearning/doc/LectureNotes/_build/html/intro.html.

[4] F. G. Gravili, *MultiVariate Analysis Tutorial*, INFN Section of Lecce and Università del Salento, Workshop ATLAS Italia, 27 October 2017 (2017), URL https://agenda.infn.it/event/13733/contributions/20520/attachments/14642/16541/MVATutorial.pdf.

[5] K. Albertsson, S. Gleyzer, A. Höcker, L. Moneta, P. Speckmayer, J. Stelzer, J. Therhaag, E. von Toerne, H. Voss, and S. Wunsch, *TMVA Users Guide*, CERN and ROOT Team, TMVA v4.3.0 for ROOT 6 (2020), URL https://root.cern.ch/download/doc/tmva/TMVAUsersGuide.pdf.

[6] TMVA Collaboration / Top Workshop, *TMVA Introduction and Tutorial (Top Workshop Talk)*, https://indico.in2p3.fr/event/305/contributions/26235/attachments/21060/25831/TMVA_TopWS_19oct07.pdf (2007), presentation at the Top Workshop, October 19, 2007.

[7] ROOT Team, *RooFit Strasbourg Tutorial (v10)*, CERN, Strasbourg, France (2025), rooFit tutorial document, URL https://root.cern/download/roofit-strasbourg-v10.pdf.